⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　昭61－84735

⑤1Int.Cl.⁴	識別記号	庁内整理番号	⑬公開　昭和61年(1986)4月30日
G 06 F 7/00	102	7313－5B	

審査請求　未請求　発明の数 1　（全11頁）

⑤4発明の名称　汎用レジスタの有効長拡張装置

⑳特　願　昭59－206235
㉒出　願　昭59(1984)10月3日

⑦2発明者　後藤　志津雄　国分寺市東恋ケ窪1丁目280番地　株式会社日立製作所中央研究所内
⑦2発明者　鍵政　豊彦　国分寺市東恋ケ窪1丁目280番地　株式会社日立製作所中央研究所内
⑦2発明者　吉住　誠一　国分寺市東恋ケ窪1丁目280番地　株式会社日立製作所中央研究所内
⑦2発明者　新谷　洋一　国分寺市東恋ケ窪1丁目280番地　株式会社日立製作所中央研究所内
⑦1出願人　株式会社日立製作所　東京都千代田区神田駿河台4丁目6番地
⑦4代理人　弁理士　高橋　明夫　外1名

明　細　書

発明の名称　汎用レジスタの有効長拡張装置

特許請求の範囲

汎用レジスタを基本とする命令体系を有する情報処理装置において、夫々の汎用レジスタの有効長を指示する有効長指示モード記憶手段と、汎用レジスタを扱う命令の動作時に汎用レジスタから読み出す情報の長さ、汎用レジスタに書込む情報の長さ、あるいは汎用レジスタを用いる演算の対象とする情報の長さを、上記のモード記憶手段に記憶された値に従つて変更する手段とを有することを特徴とする汎用レジスタの有効長拡張装置。

発明の詳細な説明

〔発明の利用分野〕

本発明は汎用レジスタを基本とする情報処理装置に係り、特に汎用レジスタの従来の長さを超えるようなアドレスやデータを取り扱うことができるような汎用レジスタの有効長の拡張を、命令体系の大巾な変更を伴なわずに実現するのに好適な汎用レジスタの有効長拡張装置に関する。

〔発明の背景〕

情報処理装置の処理するデータ量の増大に伴つて、情報の処理能力を増大させようという要請が生じる。その1つとしてアドレスの上限値の拡張、すなわち、プログラムからアクセスできるデータの量を増大（アドレス拡張と呼ぶ）させようという要請がある。他の1つは、演算処理単位の増大（データ拡張と呼ぶ）であり、1命令で処理する情報の処理単位を増加させようという要請である。これらの要請を満たす情報処理装置の作り方として、次の2つがある。1つは新しい命令体系を持つ情報処理装置を作る方法であり、他の1つは従来の情報処理装置に若干の機能追加をする方法である。ところで1つの命令体系の下で数多くのプログラムが既に作成されている場合、それらが新しい情報処理装置においてもそのまま動作しうることも要請される。この点を考えると上記の中では後者の方法、すなわち従来の命令体系に若干の修正をし、従来のプログラムを変更することなく動作させることができ、かつ新しいプログラムも

動作させることができる情報処理装置を作る方が良策である。

従来プログラムをそのまま動作させることができ、かつアドレス拡張を実現した方式としてはＩＢＭ社の「IBM System／370 Extended Architecture Pvinciples of Operations」（ＳＡ－２２－7085－０）に示されているような方式がある。そこではプログラムの実行を制御するためのプログラム状態語（ＰＳＷ）の中にアドレス拡張ビットを設け、その値が０の時にはアドレスの上限値は$2^{24}-1$，その値が１の時にはアドレスの上限値は$2^{31}-1$となる。以下、本方式をさらに詳細に説明する。第１図は、情報処理装置の命令形式の分類を表わしたものである。図において、R_1，R_2，R_3，B_1，B_2，X_2はすべて、１６個ある３２ビットの汎用レジスタの番号を指示するために用いられている。この命令体系では汎用レジスタの番号を指示するのに４ビットが使われており、１６個の汎用レジスタの中のどれかを指し示すのに適した命令体系となつている。汎用レジスタの値はアドレスであつたり、数を表現したり、あるいは文字コードであつたりするが、汎用レジスタにはそれが何を表現しているかの区別はなく、命令語の中のどの位置で指示されるかにより、その用途が決定される。すなわち、第１図において、B_1，B_2は、ベース・レジスタとしての使われ方であり、B_1，B_2の指す汎用レジスタの値とD_1，D_2の１２ビットの内容が加算された値がメモリ上のアドレスを指すのに用いられる。ＲＸ形式におけるX_2はインデックス・レジスタとしての使われ方であり、X_2の指す汎用レジスタの値およびB_2の指す汎用レジスタの値およびD_2の内容の３者の加算値がメモリ上のアドレスを指すのに用いられる。R_1，R_2，R_3はその指す汎用レジスタの値をアドレスとして用いたり、アドレスではないデータとして用いたりするが、その用いられ方は命令語によつて一意に決められている。ところで汎用レジスタの大きさは３２ビットであり、従来アドレス・モードの時には下位の２４ビットが有効となり上位８ビットの値は無視されるような命令体系であつた。汎用レジスタへのアドレスの設定は、メモリ上の３２ビットのデータを汎用レジスタにセットするLOAD命令や、３２ビットの汎用レジスタのデータとメモリ上の３２ビットのデータとを加算し汎用レジスタにセットするＡＤＤ命命、などを用いて行つている。すなわち、３２ビットのデータを基本とする命令体系を利用してアドレスの設定・計算を行つており、アドレスとして利用する時には上位８ビットを無視する操作を行つている。

このような命令体系の下でアドレスを拡張する場合を考える。１つは３２ビット以内のある長さに拡張する場合であり、他の１つは３２ビットを超えた長さに拡張する場合である。前述のIBM System／370 Extendet Architecture ではアドレスを３１ビットに拡張しているが、上述のように拡張アドレスモードか否かを表わす拡張アドレス・ビットの値に従つて３２ビットの汎用レジスタのうち上位８ビットを無視するが上位１ビットを無視するかを切換えることにより、従来プログラムと拡張プログラムの両者を１つの情報処理装置の上で実行することが可能となる。

しかし、他の１つの拡張（例えば６４ビットに拡張）を上記の方式により実現することはできない。上記の方式の拡大適用の例として、汎用レジスタの長さを６４ビットに拡大し、ベース・レジスタなどアドレスとして用いられる時には６４ビットの計算をする方式が考えられる。しかし、上記のような命令体系の下では演算は３２ビットのため６４ビットのアドレス・データの設定や計算ができない。そのため４ビットのアドレス・データの設定と計算のための命令群が新たに必要となるが、これは命令体系の大巾な変更が必要となり、好ましくない。また、拡張アドレスの設定と計算を従来の命令体系の下で実施可能とする方式として、拡張アドレス・ビットの値によつて従来の場合は３２ビットの演算を行い、拡張の場合は６４ビットの演算を行うようにする方式も考えられる。しかし、この方式を適用すると、従来の３２ビットのデータの演算が拡張アドレス・モードの時に

はできなくなるという問題が生じる。

以上、アドレスの長さの拡張を、従来の汎用レジスタの長さを超えて実施する上での問題点について説明した。

アドレスの拡張と並び要請されるのがデータ演算の単位の拡張（データ拡張）である。汎用レジスタの長さが３２ビットの従来の命令体系の下では３２ビットを超える（例えば６４ビット）の演算を行う命令はなかつた。従来、浮動小数点レジスタについては、３２ビット，６４ビット，128ビットの演算をそれぞれ別々の命令として設けることにより実施可能であつたが、汎用レジスタを使う演算（固定小数点データの演算）については、連続番号のレジスタを同時にメモリから／への転送することを除いてできず、６４ビットの固定小数点の演算を行うには３２ビットのための演算命令を多数用いて実施しているのが現状である。例えばFERTRAN などの高級言語において、処理すべきデータ量の増大に伴つて６４ビットの固定小数点データを扱う要請がある（例えば、配列の大きさが2^{32}を超えるものが要求されている。）が、６４ビットの演算のためにはコンパイラは多数の命令語に翻訳しなければならず処理を複雑にするとともに、処理性能を悪化させる要因となる。

以上、アドレス拡張とデータ拡張を従来の汎用レジスタの長さを超える程度に拡張する場合の問題について述べた。

〔発明の目的〕

本発明の目的は、命令体系の変更を殆ど行わずに、汎用レジスタの有効長を拡張し、従来のデータ長よりも大きいようなアドレスやデータを取り扱えるような、従来装置に対して変更の少ない情報処理装置を提供することにある。

〔発明の概要〕

汎用レジスタを基本とする命令体系の下でアドレス拡張やデータ拡張を行うには、汎用レジスタの長さを増大する必要がある。従来プログラムはそのまま実行でき、かつ新規拡張プログラムでは拡張されたアドレスやデータを扱う時のみ拡張された汎用レジスタを用いて拡張アドレスの計算や拡張データの演算を行い、拡張されていないデータの演算は従来と同じ長さで実施する必要がある。このためには、個々の汎用レジスタに対応して、それが拡張されたアドレス計算やデータ演算を行うか否かを表わすモード・ビットを設けるのが良い。そして、当該モード・ビットの値に従つてアドレス計算や演算の有効長を切替えることができるアドレス計算回路，演算処理回路を設け、汎用レジスタの保持している情報の長さに応じた計算や演算を実施できるようにする。

また、汎用レジスタの個数は有限であり、プログラムの実行に従つて個々の汎用レジスタの内容は変化していく。例えば最初は従来の長さのデータを保持し３２ビットの演算のために用いられる汎用レジスタが、次には拡張された長さのアドレスを保持し、拡張されたメモリ範囲内のデータをアクセスするために使用され、さらに次には、拡張された長さのデータの演算を使用されるということがありうる。このような使用方法の変更をプログラム内で実行の途中で行えるように、当該拡張ビットの変更ができるようにする必要がある。

また、マルチタスク処理を行うオペレーティング・システムの下では、複数のプログラムの間で処理を切替えるのが普通であり、汎用レジスタは切替えの際に状態が退避回復されるので、当該拡張モード・ビットをオペレーティング・システムによる退避・回復ができるようにする必要がある。

なお、このモード・ビットは従来の命令によつて変更されるようにする必要があるので、汎用レジスタの外に別に設けるのがよい。

〔発明の実施例〕

以下、本発明の一実施例を第２図より第９図を用いて説明する。

第２図は、本発明が適用される情報処理装置の構成を示す。それは次のものから構成される。

(1) 命令解析・制御部１００

命令語を解読するとともに、他の部分の処理を制御する。

(2) 命令実行部２００

命令語の実行を担当する部分であり、本発明

が適用される部分である。

(3) メモリ制御部３００

主記憶装置（ＭＳ）４００とのデータ転送を制御する。

以上の構成は一般的なものである。

次に本発明が適用される命令実行部２００内の構成を記す。

(1) 汎用レジスタ制御部１０

１６本の汎用レジスタ１１からの読出し、およびそれらへの書込みを制御する部分である。

(2) アドレス計算部２０

メモリ・アドレスを計算する部分である。

(3) 演算処理部３０

メモリから読込んだデータ，汎用レジスタから読込んだデータに対する演算を実行する部分である。

(4) モード・レジスタ制御部４０

汎用レジスタの有効長を指示するモード・レジスタ４１を制御する部分である。

上記のうち、汎用レジスタ制御部１０，アドレス計算部２０，演算制御部３０は、従来より設けられている部分であるが、モード・レジスタ制御部４０は新規な部分である。これら４部分の詳細については後述することとし、ここでは各部の動作の概要を説明する。そのために、代表的な命令として、LOAD命令とＡＤＤ命令の処理の流れを第３図を用いて説明する。命令の実行は通常、ステージと呼ばれる時間インタベルに分けて行われる。ここでは４つのステージに分ける場合を説明するが、本発明はこれに限定されるものではない。ステージごとに次のように処理される。

(1) 命令デコード・ステージ(Ｄ)

命令語内のオペレーション・コードを解読し、LOAD命令，ＡＤＤ命令などを認識するとともに、命令語内の各パートを取り出す。この処理は命令解析・制御部１００で実施されるが、このための回路構成は公知であるので説明は省略する。

(2) アドレス計算ステージ(Ａ)

命令語内のB_2，X_2，D_2パートを基に、第２オペランドのアドレスを計算する。具体的には命令解析・制御部１００の制御の下で、命令実行部２００で計算が行なわれる。すなわち、次のような処理が行われる。

(a) 命令語内のB_2パートの内容を汎用レジスタ制御部１０およびモード・レジスタ制御部に伝え、B_2の指示する汎用レジスタおよびモード・ビットをそれぞれ読み出し、さらに、アドレス計算部２０内のB_2レジスタ・データ２６およびB_2モード・ビット１２４に設定する。

(b) (ａ)と同様にして、X_2パートに対応して、X_2レジスタ・データ２７、およびX_2モード・ビット１２５に値を設定する。

(c) 命令語内のD_2パートの内容をアドレス計算部２０内のD_2データ・レジスタ２８に設定する。

(d) B_2レジスタ・データ２６，X_2レジスタ・データ２７，D_2データ・レジスタ２８の３者を入力とするアドレス・アダー２５により加算が行われ、結果がアドレス・レジスタ１２３に設定される。この際にB_2モード・ビット１２４，X_2モード・ビット１２５は、有効長の制御に使用されるが、詳細は後述する。

(3) オペランド・ロード・ステージ(Ｌ)

第２オペランドのデータをメモリからフエッチする。具体的には、メモリ・レジスタ１２３の内容がメモリ制御部３００に転送され、ＭＡ４００に対するフエッチ要求が発行される。ＭＡ４００からのフエッチが行われるとフエッチ・データ・レジスタ３０２にデータが設定される。なお、この過程においては、仮想アドレス機構のある場合にはアドレス・レジスタ123の内容は仮想アドレスを表現しており、それを実アドレスに変換する回路が動作する。また、主記憶装置ＭＳ上のデータの写しを格納する高速のバツフア記憶装置ＢＳを持つ場合には、ＢＳをアクセスする回路が動作する。本実施例にはこれらについての説明は省略しているが、本発明の適用においてはこれらの回路の有無は

無関係である。

(4) 命令実行ステージ(D)

LOAD命令の場合には、フエツチしたデータをR_1パートで指す汎用レジスタに設定する。

ＡＤＤ命令の場合には、R_1レジスタの内容とフエツチしたデータとの加算が行われ、結果をR_1レジスタに設定する。この処理は次のように行われる。

(a) 命令解析・制御部１００で取り出されたR_1パートの内容が汎用レジスタ制御部１０およびモード・レジスタ制御部４０に伝達される。

(b) モード・レジスタ制御部４０ではR_1パートの値に従つてモード・ビツトが読み出される。

(c) 演算処理部３０内のメモリ・フエツチ・データ・レジスタ３１に、フエツチしたデータ３０２の内容が転送される。LOAD命令の場合には演算器３２をバイパスして結果レジスタ１３７にデータが転送される。ＡＤＤ命令の場合には汎用レジスタ制御部１０内でR_1パートの指す汎用レジスタの内容が読出され、レジスタ出力データ・レジスタ１３１に転送され、フエツチ・データ・レジスタ３１とともにＡＬＵ３２の入力となり、加算が行われる。加算結果は結果レジスタ１３７に設定される。(b)で読み出されたR_1モード・ビツト１３６は、結果レジスタ１３７の有効長制御に用いられるが、これについては後述する。

(d) LOAD命令、ＡＤＤ命令ともに、結果レジスタ１３７の内容が汎用レジスタ制御部１０に転送され、レジスタ入力レジスタ１２を介してR_1パートの値で指定される汎用レジスタに設定される。

以上のようにして、LOAD命令、ＡＤＤ命令の処理が実行される。以上の動作は、モード・ビツトに関係する記述を除いて公知のものである。

次に、本発明の適用時の動作をそのための回路について詳細に説明する。まず、本実施例の動作の詳細を、第４図を用いて、LOAD命令について説明する。

本命令１は、X_2の指す汎用レジスタの値、B_2の指す汎用レジスタの値、およびD_2の１２ビツトの値の３者の加算値をメモリのアドレスとしてデータを読み込み、そのデータをR_1で指す汎用レジスタにセツトする命令である。従来モードにおいては、アドレスおよびデータの長さは４バイト（＝３２ビツト）以下である。本発明を実施した場合にはモード・レジスタ４１内のB_2に対応するモード・ビツトは１，X_2に対応するモード・ビツトは０とすると、B_2の指す汎用レジスタは６４ビツトの値、X_2の指す汎用レジスタは下位３２ビツトに上位３２ビツトを０と見なした値が使用される。すなわち、モード・ビツトが１の場合は対応する汎用レジスタの長さは６４ビツトとして扱い、モード・ビツトが０の場合は対応する汎用レジスタの長さは３２ビツトとして扱われる。ここで、本実施例では、拡張モードでの長さを６４ビツトとしたが、任意のビツト長さでよいことを付記する。ただし、メモリとのアクセス単位がバイト（＝８ビツト）の場合には、８の倍数に設定するのがよい。またアドレス拡張をデータ拡張を別個のモード・ビツトとして、各汎用レジスタに２ビツトを対応させたモード・レジスタの構成としうることも付記する。ここでは、データの拡張の要領の説明で述べたように８バイトの固定小数点データの実現とあわせてアドレス拡張を行うため、６４ビツトとした。またデータ拡張としては６４ビツト・アドレス拡張としては（２４ビツト・アドレス方式のように）その一部（例えば下位４８ビツト）をアドレスとして使用する方式も同様にして実現できることを付記する。

第４図において、B_2，３，X_2，４，D_2，５の３者の和がメモリ上のデータにアクセスする６４ビツトのアドレス６として使用される。メモリからの読出しデータの長さはR_1に対応するモード・ビツトに依る。モード・ビツトが１の時は、R_1の指す汎用レジスタの６４ビツトに８バイトのデータ（Ⓐ，Ⓑの部分）が読出されてセツ

トされる。モード・ビットが0の時は、4バイトのデータ（Ⓐの部分）が読出され、Ⓐの最上位ビットxの値が0の時その上位に32ビットの0が付加された値が、またxが7の時1が付加された値が、それぞれ汎用レジスタ7にセットされる。

次に、本実施例の詳細な説明を行う。まず、汎用レジスタ制御部10の動作を第5図により説明する。汎用レジスタは16本あり（11）それらとの入出力はレジスタ番号指定レジスタ14により選択して実行される。すなわち、レジスタ番号指定レジスタ14の内容に従つて16本の汎用レジスタ11の中の1つが選択され、書込みの場合はレジスタ入力データ・レジスタ12を経由して、また読出しの場合はレジスタ出力データ・レジスタ13を経由して64ビット単位での書込み／読出しが実施される。

次に、アドレス計算部20の動作を第6図を用いて説明する。第4図に述べたLOAD命令1の例について説明する。命令語によつて指定されたレジスタX_2の内容が第5図の汎用レジスタ制御部10から読出され、X_2レジスタ・データ・レジスタ21にセットされる。またX_2レジスタに対応するモード・ビット125がモード・レジスタ制御部40から読出される（詳細は後述）。X_2モード・ビット125の値が0の時には、0を保持している定数レジスタ22の内容がアドレス・アダー25のX_2入力レジスタ26の前半32ビット・セットされる。X_2モード・ビット125の値が1の時には、X_2レジスタ・データ・レジスタ21の前半32ビットがX_2入力レジスタ26の前半32ビットにセットされる。どちらの場合も後半32ビットにはX_2レジスタ・データ・レジスタ21の後半32ビットの値がそのままセットされる。B_2についてもX_2について説明した上記の処理が同様にして行われ、B_2入力レジスタ27にセットされる。命令語1のD_2パートはそのままD_2入力レジスタ28にセットされる。X_2入力レジスタ26，B_2入力レジスタ27，D_2入力レジスタ28はアドレス・アダー25の3入力となり、3者の加算が実施され、結果が出力レジスタ121にセットされる。選択器122は、X_2モード・ビット125およびB_2モード・ビット124がともに0の時には、アドレス・レジスタ123の前半32ビットには定数レジスタ22の内容をセットし、少くとも一方の1の場合には出力レジスタの前半32ビットの内容をセットするために用いられる。どちらの場合においてもアドレス・レジスタ123の後半32ビットは、出力レジスタ121の後半32ビットがそのままセットされる。以上の結果、X_2レジスタ、B_2レジスタの両者が従来モードの時には有効長32ビットのアドレスを生成し、少くとも一方が拡張モードの時には有効長64ビットのアドレスが生成される。生成されたアドレスに従い、メモリ・リクエストが実行されるが、その動作回路は公知なので省略する。

次に、演算制御部30の説明を第7図を用いて行う。先ず、LOAD命令の処理の続きを説明する。上述のメモリ・リクエストの結果として、メモリから読込まれたデータがメモリ・フエッチ・データ・レジスタ31にセットされる。その内容は、64ビットのALU32の入力レジスタ33にセットされるとともに、選択器34の入力となる。LOAD命令の場合には、選択器34によりメモリ・フエッチ・データがそのままモード選択器35の入力となる。メモリ・フエッチ・データ・レジスタ31の前半部はさらに、32ビットのALU36の入力レジスタ37の入力となるとともに、選択器38の入力となる。LOAD命令の場合は上と同様にメモリ・フエッチ・データがそのままデータ拡張器39の入力となり、さらにデータ拡張器39により64ビットのデータに拡張される。すなわち、32ビットのデータの最上位1ビットの値が1の時にはすべて1の32ビットのデータが付加され、0の時にはすべて0の32ビットのデータが付加される。このための回路も既知なのでここでは説明は省略する。データ拡張器39の出力は、前述のモード選択器35の入力となる。モード選択器35においては、命令語のR_1の指すレジス

タに対応するモード・ビットがモード・レジスタ制御部４０より伝えられ、その値が１の時には、６４ビット系のデータを、その値が０の時には３２ビット系のデータを、それぞれ選択し、結果１３７をレジスタ入力データとして汎用レジスタ制御部１０に転送する。汎用レジスタ制御部１０では既述の如く汎用レジスタ１１にセットする。以上のようにしてLOAD命令の処理（第４図で述べた処理）が完了される。次に、データが拡張の例としてＡＤＤ命令の処理を本図を用いて説明する。本命令はLOAD命令１と同じＲＸ形式であり、X_2，B_2，D_2により指示されるメモリ上のデータをフエッチし、R_1の指す汎用レジスタの内容と加算し、結果を元のR_1の指す汎用レジスタにセットする命令である。メモリからフエッチされたデータはLOAD命令の時と同じ処理によりメモリ・フエッチ・データ・レジスタ３１にセットされる。ＡＤＤ命令の場合には、６４ビットＡＬＵ３２と３２ビットＡＬＵ３６の演算が実施される。両ＡＬＵの入力には、汎用レジスタ制御部１０により読出されたデータがレジスタ・データ・レジスタ１３１にセットされる。その全体６４ビットは６４ビットＡＬＵの入力レジスタ１３２に、下位３２ビットは３２ビットＡＬＵの入力レジスタ１３３にセットされる。ＡＤＤ命令の場合には、LOAD命令のときとは異なり選択器３４，３８ではＡＬＵ出力１３４，１３５が選択される。後者はデータ拡張器３９により６４ビットのデータに拡張される。選択器３４の出力とデータ拡張器３９の出力はモード選択器３５の入力となり、R_1モード・ビットの値に従つてどちらかが選択され、レジスタ入力データ１３７となる。この値１３７は汎用レジスタ制御部１０において、R_1の指す汎用レジスタにセットされる。このようにして、ＡＤＤ命令が実行される。ここで、６４ビットＡＬＵ３２は本発明で新たに設けたものであるが、３２ビットＡＬＵを容易に拡張して実現できるので、詳細は省略するが、演算の長さが３２ビットから６４ビットになつただけであり、オーバフローなどの割込み要因発生回路も３２ビットＡＬＵと同様に先頭ビットからのオーバフローの検出などを行なえばできる。また、本図の６４ビットＡＬＵ，３２ビットＡＬＵ，データ拡張器などを一体とした同じ機能をもつＡＬＵを作成することも容易にできる。

次に、モード・レジスタ制御部の動作を第８図を用いて行う。命令語１のレジスタパート１０７がレジスタ番号４４に設定され、セレクタ４２のセレクタ信号となる。モード・レジスタ（ＭＲ）４１はセレクタ４２の入力となり、指定されたレジスタ番号に対応する部分がモード・ビット110として出力され、上述したアドレス計算部２０、演算処理部３０に伝達される。

次にモード・レジスタ４１の設定，変更，メモリへの転送のために設ける命令について第８図，第９図を用いて説明する。４つの命令を新たに設けるのが良い。４つの命令はすべて第１図の中のＳ形式である。Ｓ形式のアドレス指定部（B_2，D_2）は、メモリ・アドレスを指したり、後述のＳＲ演算，ＡＮＤ演算のデータとして用いられる。

(1) OR Mode Register（ＯＭＲ）命令

命令語で指定されたオペランドアドレスのうち下位１６ビットの情報とモード・レジスタ４１の情報の各ビット対応のＯＲ演算が行われ（第９図のステップ２０１）結果がモード・レジスタに設定される（ステップ２０２）。すなわち、第８図において、デコーダ４３によりＮＭＲ命令信号１４１がオフになる。ところで、MR41の内容はレジスタ１４２を経由してＡＮＤ回路１４３，ＯＲ回路１４４の入力となる。また、オペランド・アドレスはレジスタ１４５を経由して両回路の入力となり、両者のＡＮＤ演算，ＯＲ演算が行われ、結果はレジスタ１４５，１４６にそれぞれセットされる。上述のようにＯＭＲ命令の場合、ＮＭＲ命令信号線はオフになつておりＯＲ回路の出力１４６が選択される。さらにＬＭＲ命令信号線１４７はオフとなり、ＭＲ書き込み信号線１４８はオンとなるようにデコードされ、ＯＲ回路の出力１４６がＭＲ41にセットされる。

(2) AND Mode Register（ＮＭＲ）命令

ＯＭＲ命令において、ＯＲ演算がＡＮＤ演算になる他は同じであり、ＮＭＲ命令信号線141がオンとなるようにデコードされ、ＡＮＤの結果１４５がＭＲ４１にセットされる。

(3) Load Mode Register（ＬＭＲ）命令

命令語で指定したアドレスのデータ２バイトをメモリより読込み（ステップ２０５）、モード・レジスタ４１に設定する(ステップ２０６)。すなわち、メモリ・フエッチ・データ・レジスタ１４９の上位１６ビットをＭＲ４１に書込む。デコーダ４３において、ＬＭＲ命令信号線147がオンになり、メモリ・フエッチ・データ・レジスタ１４９の入力を選択し、ＭＲ書込み信号線１４８をオンになることによりＭＲ４１に書込まれる。

(4) Store Mode Register (STMR) 命令

モード・レジスタ４１の内容が、本命令語で示されたアドレスにストアされる。本命令の場合、デコーダ４３によりSTMR命令信号線１５０がオンになり、メモリ・ストア・データ・レジスタ１５１を経由して、メモリ・リクエストが実行される。

以上、モード・レジスタ４１に関する４つの命令の動作を説明したが、これらの命令は非特権命令とした方がよい。すなわち、汎用レジスタの内容は、プログラム毎に変化し、プログラム内においても時々刻々変化するので、非特権プログラムにおいても上記の命令が使用可能とするためである。また、モード・レジスタの初期値はすべて０とするのがよい。

以上、本実施例の説明を行つた。本実施例によれば、次のような効果がある。

(1) 従来のプログラムを変更しなくても、そのまま実行することができる。モード・レジスタの初期値は０なので、従来の有効長として扱われるので、従来プログラムの変更はまつたく不要である。

(2) 従来プログラムを変更して、データ・アドレスのみ拡張アドレス部分にアクセスするようにするには、拡張アドレス部分にアクセスする前に、拡張アドレス用のレジスタとして使われる汎用レジスタに対応するモード・ビットを１にして、メモリ・アクセスを行い、拡張アドレス用のレジスタとしての使用が終了した時点でモード・ビットを０にするような変更、すなわち２命令の追加を行えばよいので容易にアドレス拡張が実現できる。

(3) 従来プログラムを、拡張アドレス部分に格納して動作させる場合においても、プログラム部分のベース・アドレスとして使われる汎用レジスタに対して、上記と同じような変更を加えるだけで済む。

(4) 固定小数点データの処理単位を従来の３２ビットから６４ビットに拡張することも上記と同様に拡張データ用に使用する前に汎用レジスタのモード・ビットを１にするだけで、従来と同じ命令体系の命令語を使用して、６４ビットの固定小数点の演算ができる。

(5) 新規に拡張アドレスで動作するプログラムや拡張データを扱うことができるプログラムを作成する際においても、従来のコンパイラなどのソフトウエア技術をほとんどそのまま活用することができる。

なお、拡張モードで使用されている汎用レジスタの内容をメモリに記憶したり、メモリから値を設定する場合には、そのための格納領域の大きさを従来の４バイトから８バイトに変更する必要が生じるが、その操作は容易である。

〔発明の効果〕

本発明によれば、命令体系の変更は殆ど行わずに、アドレス拡張やデータ拡張が実現できるので、従来プログラムの流用、および従来のコンパイラなどのソフトウエアの流分が容易になるという効果がある。すなわち、(１)命令体系が殆ど変わらず、(２)汎用レジスタが拡張アドレスまたは拡張データとして使用される時のみ、対応するモード・ビットの変更を行う命令を追加すれば、アドレス拡張ならびにデータ拡張が実現できるので、過去に蓄積された膨大なプログラムをそのまま実行

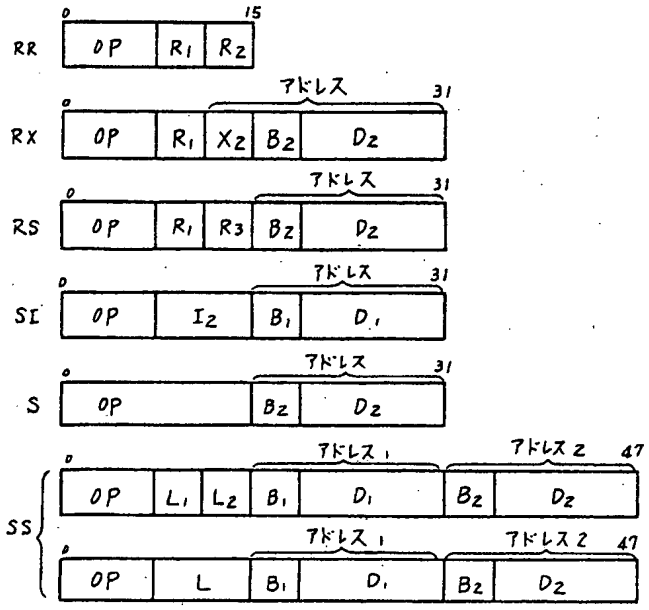
することはもちろん、若干の修正のみで拡張アドレスで動作させたり、拡張データを扱つたりすることができるようになる。以上の結果、アドレス拡張やデータ拡張に伴ない必要となるソフトウエアの開発工数を非常に小さくできるという効果がある。

図面の簡単な説明

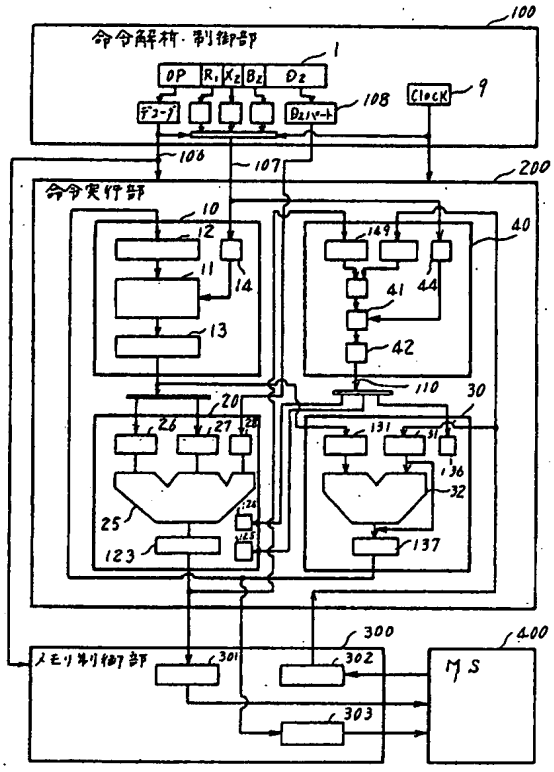
第１図は情報処理装置の命令形式の説明図、第２図は本発明の実施例の構成図、第３図はLOAD命令，ＡＤＤ命令の動作概要図、第４図はLOAD命令の詳細な動作図、第５図は汎用レジスタ制御部の回路構成図、第６図はアドレス計算部の回路構成図、第７図は演算処理部の回路構成図、第８図はモード・レジスタ制御部の回路構成図、第９図はモード・レジスタ演算命令の動作の説明図である。１０…汎用レジスタ制御部、１１…汎用レジスタ、２０…アドレス計算部、２５…アドレス・アダー、３０…演算処理部、３２，３６…ＡＬＵ、４０…モード・レジスタ制御部、４１…モード・レジスタ、２３，２４，１３４，３８…セレクタ、３５…モード選択器、３９…データ拡張器、１４３…ＡＮＤ回路、１４４…ＯＲ回路、１００…命令解析制御部、２００…命令実行部、３００…メモリ制御部、４００…主記憶装置。

代理人　弁理士　高橋明夫

第１図



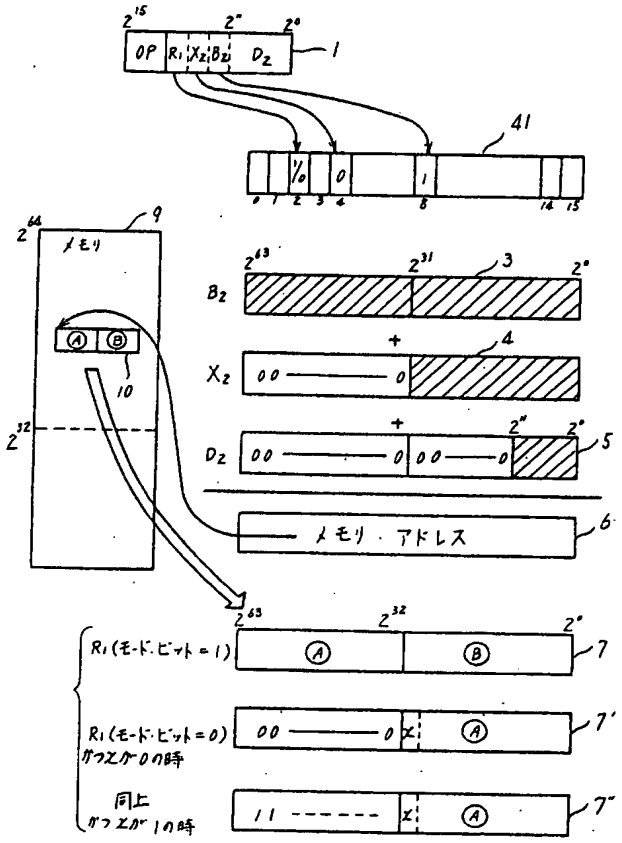
第２図



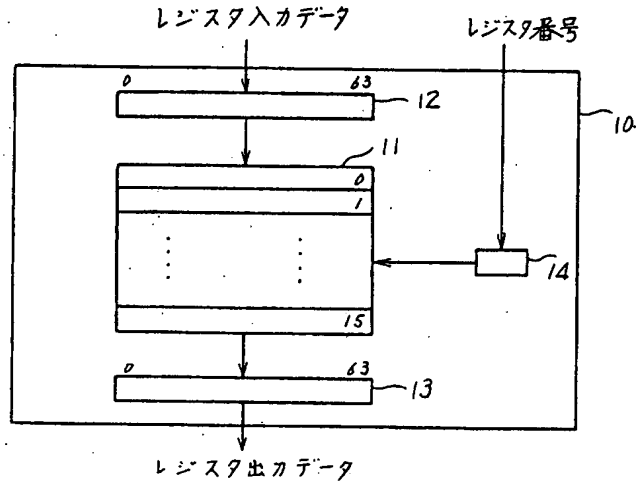
第3図

記号	ステージ名	LOAD命令	ADD命令
D	命令デコードステージ	・命令語内のオペレーション・コードによりLOAD命令と認識する。	・命令語内のオペレーション・コードによりADD命令と認識する。
A	アドレス計算ステージ	・命令語内のB_2, X_2, D_2 パートを基に、オペランドのアドレスを計算する。 ① B_2モードビットとB_2レジスタ・データを読出す。 ② X_2モードビットとX_2レジスタ・データを読出す。 ③ B_2レジスタ・データ, X_2レジスタ・データ, D_2 パートの3者の加算結果をアドレスとする。	
L	オペランドロードステージ	・オペランドのデータをメモリからフェッチする。 ① Aステージの結果のアドレスを用いて、メモリ・フェッチを行なう。 ② フェッチしたデータをメモリ・フェッチ・データにセットする。	
E	命令実行ステージ	・命令語内のR_1 パートの指すレジスタに、Lステージでフェッチしたデータをセットする。 ① R_1モードビットを読出す ② R_1モードビットの値に従いセットする長さを変更する	・命令語内のR_1パートの指すレジスタの内容とLステージでフェッチしたデータの加算を行い結果を当該レジスタにセットする。 ① 同左 ② 同左

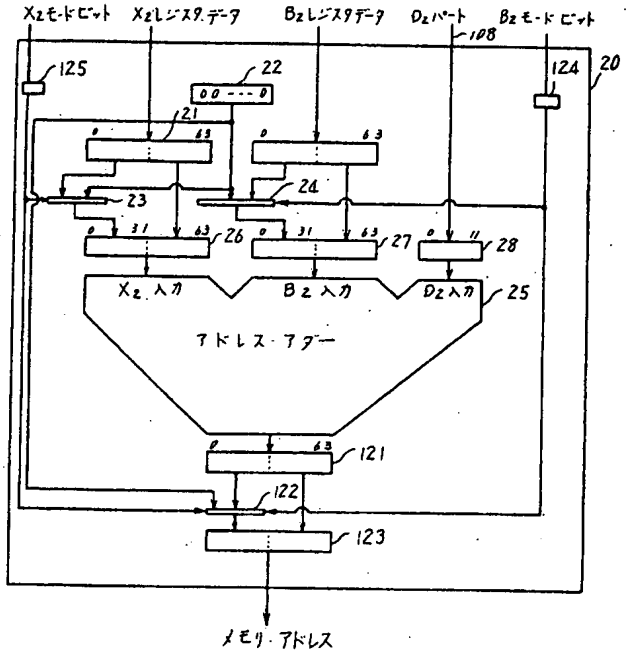
第4図



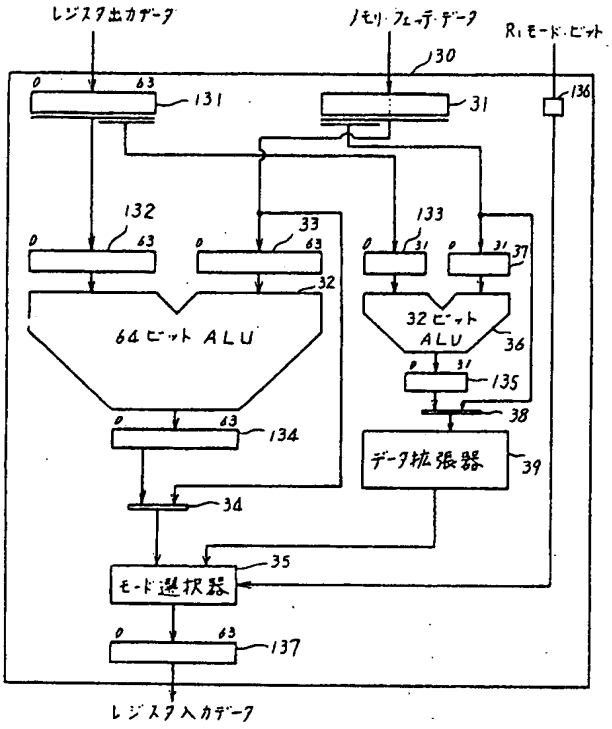
第5図



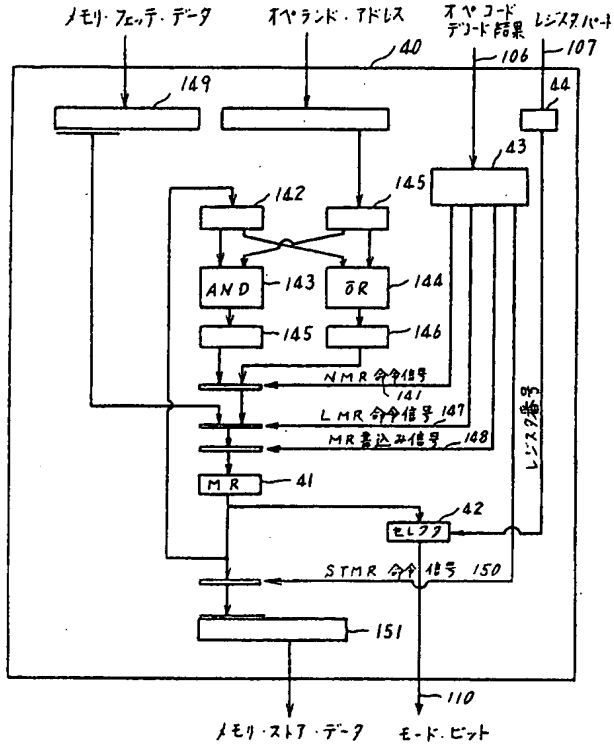
第6図



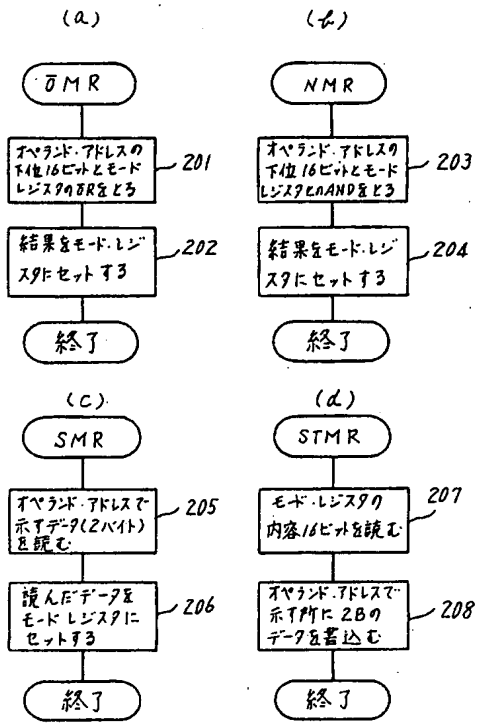
第 7 図


レジスタ出力データ
メモリ・フェッチ・データ
R1モード・ビット
30
131
31
136
132
33
133
37
64ビット ALU
32
32ビット ALU
36
135
38
134
データ拡張器
39
34
35
モード選択器
137
レジスタ入力データ

第 8 図


メモリ・フェッチ・データ
オペランド・アドレス
オペコード デコード結果
レジスタバート
40
106
107
149
44
43
142
145
AND
143
OR
144
145
146
NMR 命令信号
141
LMR 命令信号
147
MR書込み信号
148
レジスタ番号
MR
41
セレクタ
42
STMR 命令信号 150
151
110
メモリ・ストア・データ
モード・ビット

第 9 図


(a)
OMR
オペランド・アドレスの下位16ビットとモードレジスタのORをとる
201
結果をモード・レジスタにセットする
202
終了
(b)
NMR
オペランド・アドレスの下位16ビットとモードレジスタとのANDをとる
203
結果をモード・レジスタにセットする
204
終了
(c)
SMR
オペランド・アドレスで示すデータ(2バイト)を読む
205
読んだデータをモード・レジスタにセットする
206
終了
(d)
STMR
モード・レジスタの内容16ビットを読む
207
オペランド・アドレスで示す所に2Bのデータを書込む
208
終了